

TOE1G-IP FTP サーバー・リファレンス・デザイン説明書 (Xilinx 版)

Rev1.1J 2016/09/01

このドキュメントは Xilinx 製 FPGA 評価ボードで動作する TOE1G-IP コアの FTP サーバー・リファレンス・デザインに関して説明したものです。実ボードでのデモ手順については以下の手順書を参照してください。

[文書名] TOE1G-IP FTP サーバー・デモ手順書(Xilinx 版)
 [ファイル名] dg_toe1gip_instruction_xilinx_jp.pdf
 [入手先 URL] http://www.dgway.com/TOE1G-IP_X.html

1 概要

ファイル転送プロトコル(FTP)はインターネット上でファイルを共有するために作成されたプロトコルです。下位レイヤにて TCP/IP を使うことで FTP はデータの信頼性を維持しながら効率的に転送することができます。

参考技術資料 (英文資料)

1. File Transfer Protocol: <http://tools.ietf.org/html/rfc959>
2. File Transfer Protocol: http://www.tcpiptide.com/free/t_FileTransferProtocolFTP.htm
3. FTP Sequence: www.eventhelix.com/realtimemantra/networking/FTP.pdf
4. List of FTP commands: http://en.wikipedia.org/wiki/List_of_FTP_commands
5. List of FTP server return codes: http://en.wikipedia.org/wiki/List_of_FTP_server_return_codes

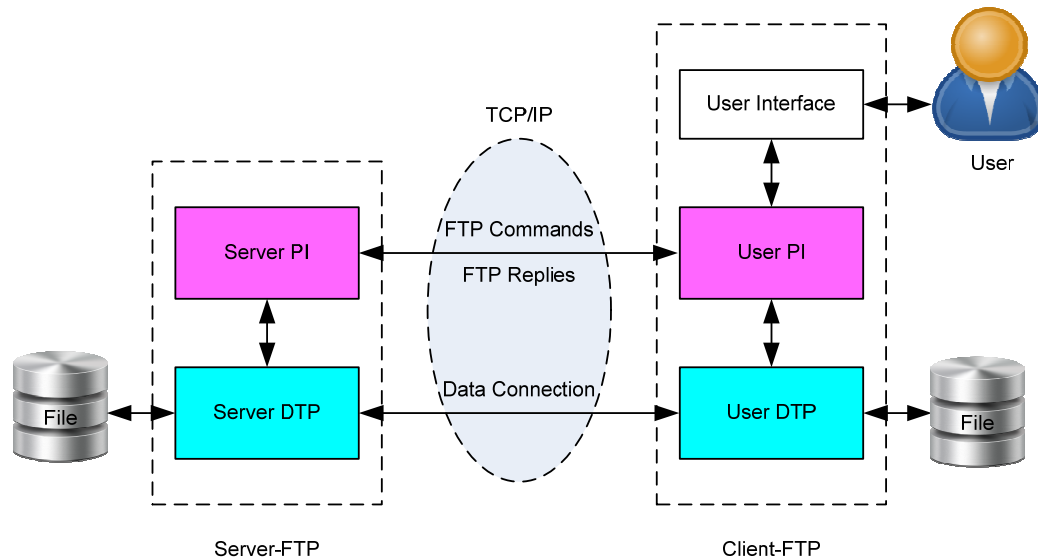


図 1: FTP モデル

FTP においては図 1 に示すように、それぞれが自身のファイルを保存するストレージ空間を持つ2種類のホストがあります。ひとつはネットワーク内でデータを共有するためのサーバーであり、もう一つは一般ユーザーがアクセスするためのクライアントです。FTP プロトコルを使ってファイルを転送するためには2つのポート接続が必要で、すなわちひとつは制御ポートで FTP コマンドをクライアントからサーバーへ転送しサーバーからクライアントに FTP 応答を転送します、そしてもうひとつはデータ・ポートでホスト間にてデータを転送します。

サーバー・プロトコル・インタープリタ(サーバーPI)およびユーザー・プロトコル・インタープリタ(ユーザーPI)はそれぞれサーバーとクライアントのコントロール接続を管理します。サーバーPI は一般的に TCP ポート番号となるポート 21 を監視(Listen)しユーザーPI からの接続要求をチェックします。接続要求によりコネクションが確立されるとサーバーPI はユーザーPI からの FTP コマンドを受信し、FTP 応答を返送し、サーバー・データの転送処理を管理できるようになります。ユーザーPI はユーザー・インターフェイスからのコマンドを処理しサーバーPI に転送し、返ってきた応答をユーザーに伝達し、そしてユーザー・データの転送処理を管理します。

サーバー・データ転送処理(サーバーDTP)およびユーザー・データ転送処理(ユーザーDTP)はデータの送信または受信に使われます。データの接続はサーバーDTP から(アクティブ・モード)あるいはユーザーDTP から(パッシブ・モード)で確立されます。一般的には FTP クライアントを保護するファイアーウォールの問題を解決するためパッシブ・モード FTP が実装されます。データ・接続はファイルをサーバーとクライアント間で転送するため確立され、ファイル転送が完了すると接続が終了します。サーバーDTP とユーザーDTP はファイルのリードやライトを行うローカルのファイル・システムに影響し合います。

ユーザー・インターフェイスは FTP ソフトウェアにより使用者となるユーザーからのコマンド実行と結果報告を提供します。

1.1 FTP 接続確立とユーザー認証

接続が接続された後に最初に行う必要がある処理は、FTP サーバーに対して許可されたユーザーのみアクセスを許可するためのユーザー認証です。接続確立とログインの詳細について以下に説明します。

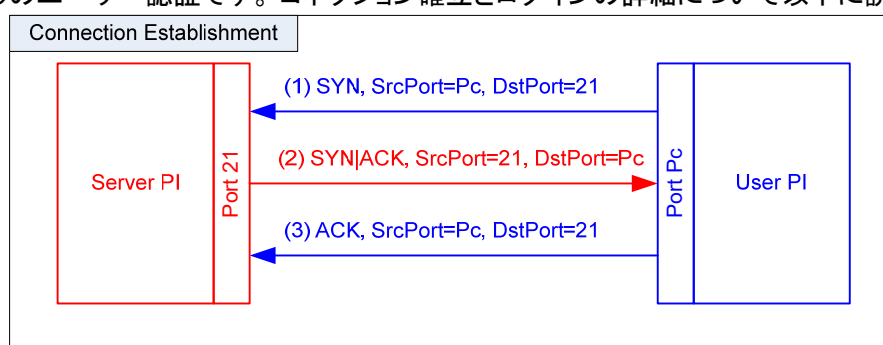


図 2:接続確立時の制御ポート

接続の確立は TCP プロトコルに準拠して図 2 に示す 3ウェイ・ハンドシェイクで行われます。FTP クライアントはサーバー側ポート 21 を指定し SYN フラグをセットした TCP パケットを送信します。FTP サーバー側ではこの新たな接続要求を検出すると接続を受諾したことを示す SYN および ACK フラグをセットして TCP パケットを返信します。クライアントは本プロセスを完了するため ACK フラグをセットし送信します。この後サーバーとクライアントは FTP コマンドやその応答で通信することができるようになります。

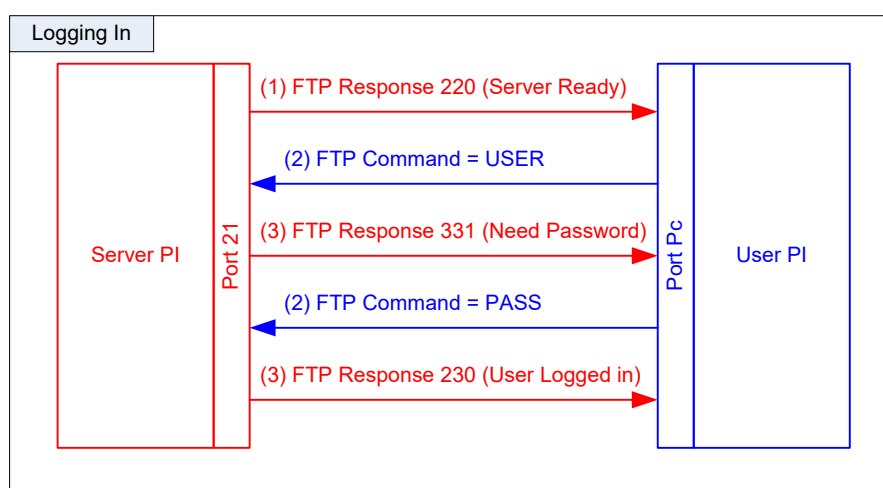


図 3: ユーザー認証

図 3 に示すように制御ポートの接続が確立するとユーザー名(USER コマンド)とパスワード(PASS コマンド)の 2 つの FTP コマンドがユーザーPI から順に送信されます。そして最後にサーバーPI は新しいセッションの開設を許可したことを意味する応答 230 をクライアントに返します。

1.2 パッシブ・モードでの FTP データ・コネクション管理

このリファレンス・デザインでは一般的に使われるクライアント(パッシブ・モード)でデータの接続が確立するよう実装されています。従ってこのトピックではパッシブ・モードでのプロトコル管理のみ説明します。

例えば LIST(サブ・ディレクトリやファイルの表示)、STOR(ファイルの格納)、RETR(ファイルの取得)などいくつかの FTP コマンドではデータ・コネクションを通じたデータ転送が必要となります。データを転送するのに先立って、サーバー-DTP にてポート番号を指定するため PASV コマンドが発行されます。

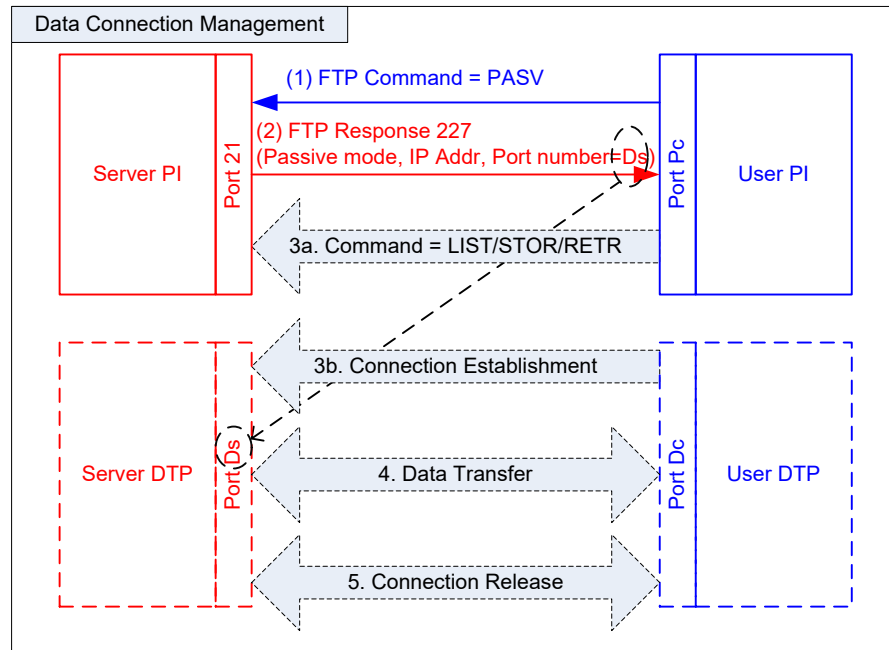


図 4: パッシブ・モードでのデータ・コネクション管理

PASV コマンドを受信するとサーバーは応答 227 を返しサーバー側の IP アドレスとデータ・ポート番号をクライアントに通知します。この例ではサーバー側のデータ・ポート版五を'Ds'としています。そしてサーバーPI はサーバーDTP に対してポート Ds をリスン状態に制御します。ユーザーPI はサーバーPI に対しデータ転送を必要とする FTP コマンドの送信とあわせてユーザーDTP はデータを転送するためにデータ・コネクションを確立します。FTP コマンドのステップ 3a とデータ・コネクション確立のステップ 3b の順序は FTP クライアントの動作によっては入れ替わることもあります。データ転送が完了するとデータ・コネクションは解放されます。

1.3 LIST コマンド

LIST コマンドは指定されたディレクトリ内のファイル・リストをデータ・コネクションを介して転送するために使われます、このため予めデータ・コネクションを初期化するのにPASVコマンドを発行する必要があります。ユーザーPIはサーバーに対してLISTコマンドを送信する一方、ユーザーDTPはデータ・コネクションを確立します。ステップ2aとステップ2bの順番は入れ替わることがあります。

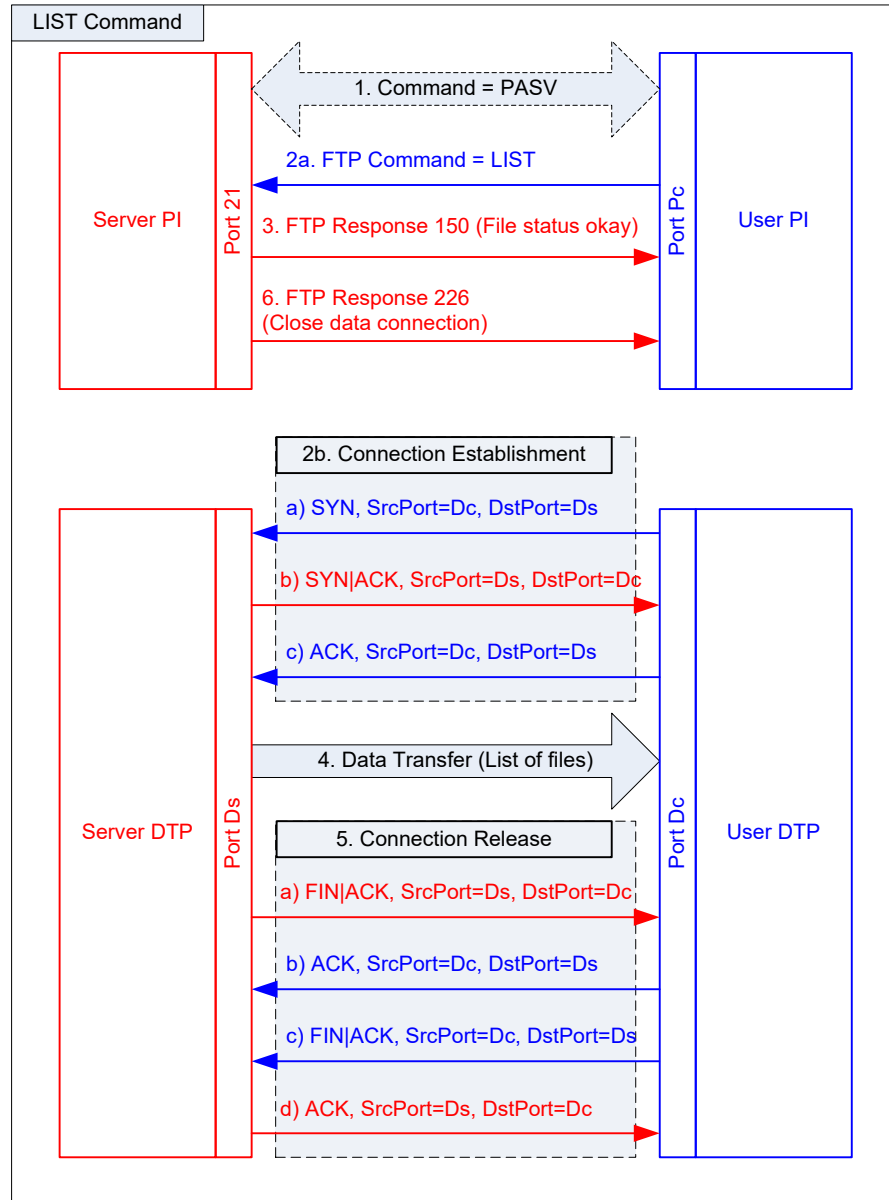


図 5: LIST コマンドのシーケンス

サーバー側ではサーバーPIはLISTコマンドに対して応答150を返送し、サーバーDTPはクライアントにファイルのリスト情報を返します。データ転送が完了すると、サーバーDTPによりデータ・コネクションは解放されます。サーバーPIから応答256が送信され一連のLISTコマンド動作を完了します。

1.4 STOR コマンド

STOR コマンドはクライアントからサーバーにファイルを保存するために使われます。図 6 に示したようにサーバー PI での STOR コマンドのシーケンスは LIST コマンドに似ています、しかしデータ・コネクションでのデータ転送の方向が逆になります。サーバーは STOR コマンドからファイル名を取得し、データ・コネクション中の総データ転送サイズから計算されたファイル・サイズを取得します。ファイル・データはクライアントからサーバーに転送されるため、データ・コネクションは転送完了時にユーザー DTP から解放されます。

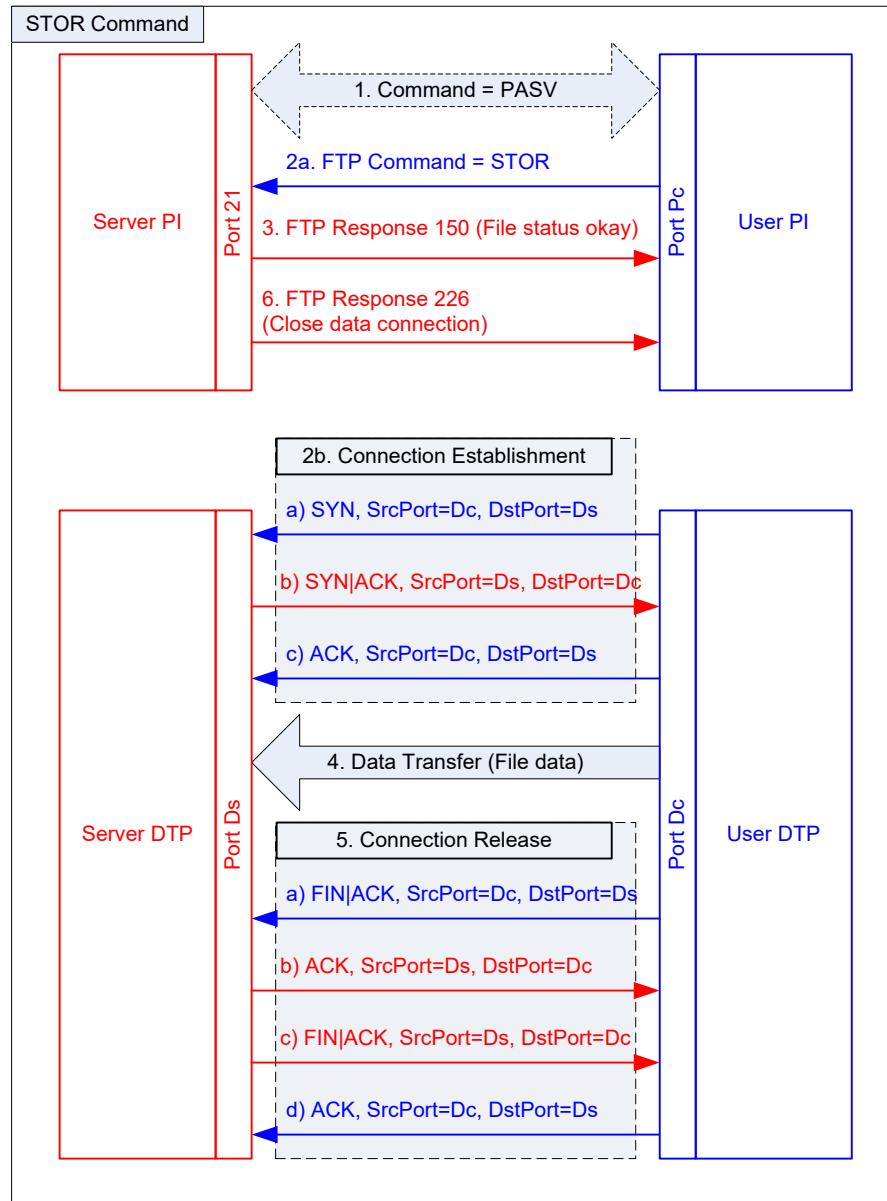


図 6: STOR コマンドのシーケンス

1.5 RETR コマンド

RETR コマンドはクライアントがサーバーからファイルを獲得するために使われます。このコマンドのシーケンスは LIST コマンドと似ています、ただサーバーから戻されるデータはファイルのリストではなくファイル・データです。STOR コマンドと同様、ファイル名は RETR コマンド・パラメータから取得されます。データ転送が完了するとサーバー DTP よりデータ・コネクションが解放されます。

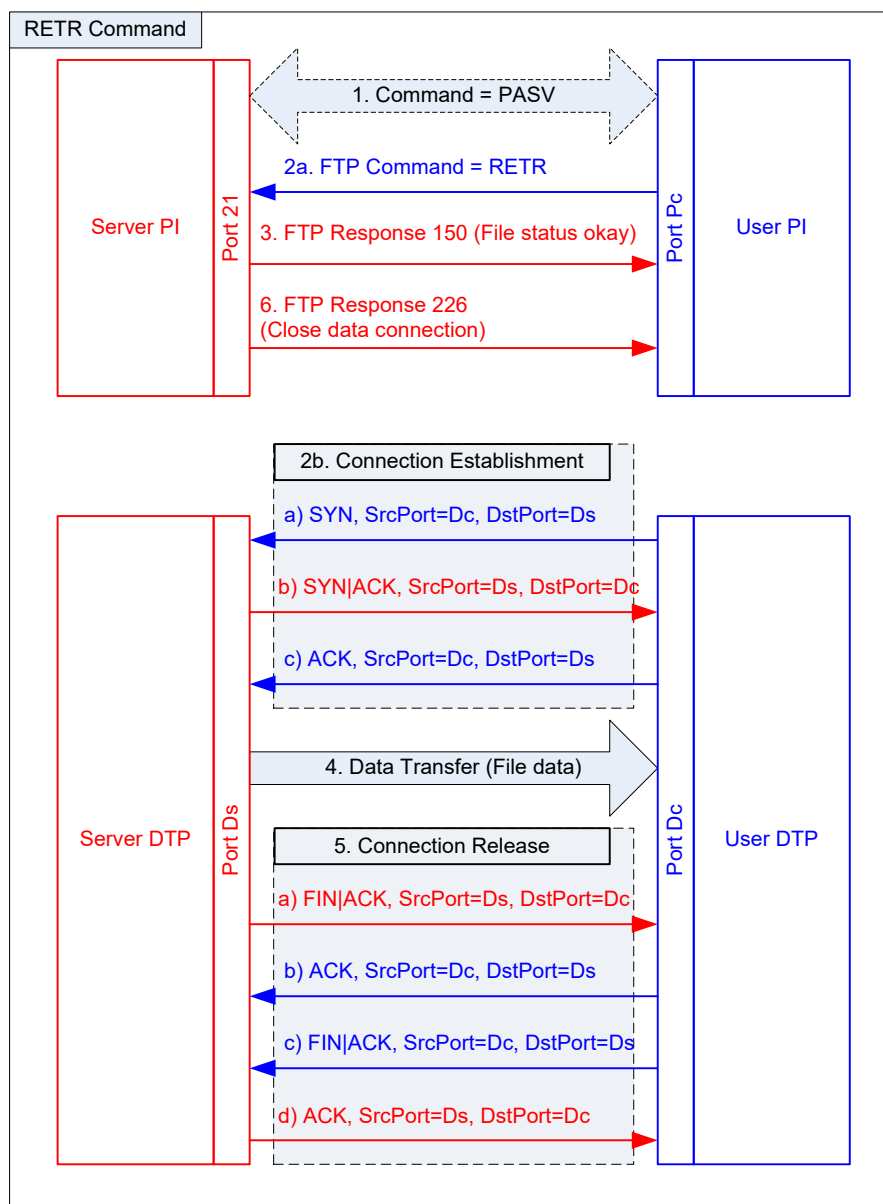


図 7: RETR コマンドのシーケンス

2 ハードウェアの構造

図 8 に示すように FTP サーバー・デモのハードウェア構成は 2 ポート・リファレンス・デザインと同一です。このためまずは 2 ポート版リファレンス・デザインの説明書を参照してください。

[文書名] TOE1G-IP 2 ポート・リファレンス・デザイン説明書(Xilinx 版)

[ファイル名] dg_toe1gip_2port_refdesign_xilinx_jp.pdf

[入手先 URL] http://www.dgway.com/TOE1G-IP_X.html

本 FTP サーバー・デモにおいては 2 ポート版デザインから CPU ファームウェアのみ編集されています。本説明書では 2 ポート版デザインより FTP サーバー・デザインで追加された箇所のみ説明します。

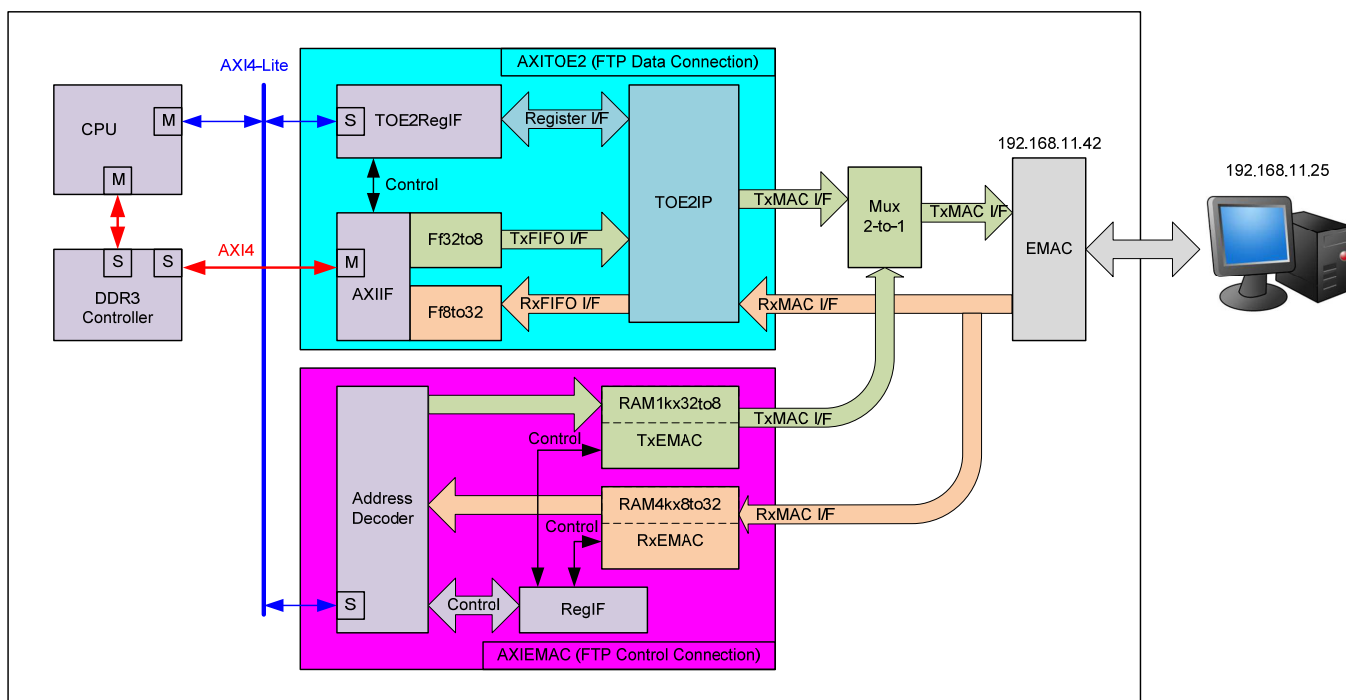


図 8: FTP サーバー・デモのハードウェア・ブロック図

本デザインは 2 ポート・デザインと比較すると高速コネクションが FTP データ・コネクションとして使われる一方、低速コネクションは FTP 制御コネクションとして使われます。FTP 制御コネクションはクライアントからの FTP コマンドを受信し FTP 応答をクライアントへ返送するために使われます。制御ポートはポート 21 に固定されています。従って CPU ファームウェアにおいては RxEMAC モジュール内のヘッダ・フィルタリング機能にて 2 ポート版デザインとは異なる値を設定し ICMP プロトコルの代わりに FTP 制御コネクションをサポートする必要があります。

FTP サーバー・デモをサポートするため、RxEMAC モジュールのヘッダ・フィルタ機能は以下のように設定されます。

- 接続先 MAC アドレス = ブロード・キャスト ID または CPU からセットされた値
- 接続先 IP アドレス = CPU からセットされた値
- プロトコル = TCP (RXPATT23_REG=0x06, RXPATTEN_REG[0]='1')
- 接続先ポート番号 = 21 (RXPATT36_REG=0x0015, RXPATTEN_REG[2:1]='11')

この設定により FTP コマンド・パケットのみがフィルタリングされ RxRAM に格納されます。

3 ソフトウェア

第1章にてFTP サーバー・デザインにおける制御およびデータ・コネクションのシーケンス詳細は説明済みです。

本デザインではメインの制御は CPU ファームウェアにより実装されています。CPU は各コネクションの制御をレジスタ・アクセスにより実行します。制御ポートのイーサネット・データは AXIEMAC モジュール内の TxRAM および RxRAM で構築/デコードされます。一方データ・ポートの TCP データは DDR3 を通じて転送します。この章では FTP サーバー・アプリケーションにおいて DDR3 のメモリ・マップを示し、またデザインで実装されている FTP コマンドと応答、ソフトウェアによるシーケンス制御について説明します。

3.1 DDR3 メモリ・マップ

FTP サーバー内のデータはファイル・システムのフォーマットで格納する必要があります。本デモでは簡易的なファイル・システムを実装するため、図 9 に示すように FTP のファイル・データのみではなくファイル情報もあわせて DDR3 メモリ内に格納します。簡易的なファイル・システムのためサブ・ディレクトリには対応せず、また格納可能なファイル数は最大 16 ファイルまでとなります。

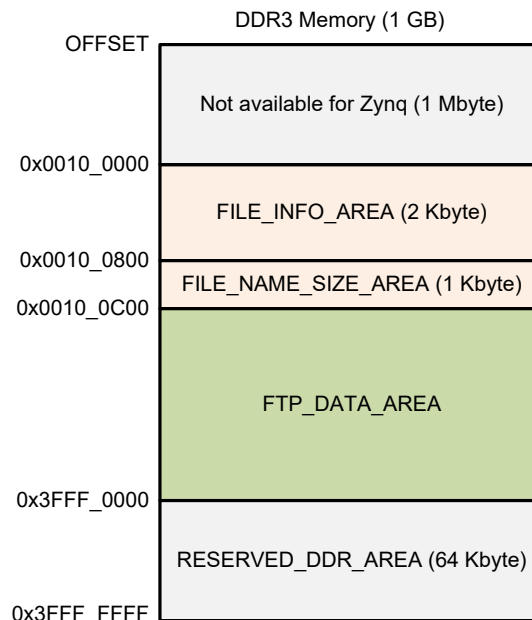


図 9: DDR3 メモリ・マップ

DDR3 は以下のように 4 つのエリアに分かれます

- 1) FILE_INFO_AREA: (ファイル情報エリア)
このエリア内のデータは"LIST"コマンド受信時に返送するためのものです。ファイル情報にはファイル名、ファイル・サイズ、ファイル権限、更新日情報が格納されます。本エリア向けに 16 ファイル分 2K バイトの空間を占有しますが実際に使われるのはこのうち 1.2K バイトです。
- 2) FILE_NAME_SIZE_AREA: (ファイル名サイズ・エリア)
このエリア内のデータは"MLSD"コマンド受信時に返送するためのものです。MLSD コマンドで返送されるデータはファイル・サイズとファイル名です。MLSD コマンドと LIST コマンドの戻りデータ・フォーマットは異なるため、それぞれ異なる DDR3 空間で用意されています。このエリアは 1K バイトを占有します。
- 3) FTP_DATA_AREA: (データ・エリア)
FTP サーバーのストレージ空間をエミュレートします。従って全ファイルの最大合計サイズは 1G バイト - 1M バイト - 67K バイトとなります。
- 4) RESERVED_DDR_AREA: (予約エリア)
このエリアはユーザーからのデータが FTP_DATA_AREA に格納しきれずオーバー・フローしてしまう場合に TOE1G-IP コア内の FIFO データをダミーで上書きフラッシュするために使います TOE1G-IP コアの最大バッファ・サイズは 64K バイトのためこのエリアは 64K バイトを占有します。

3.2 FTP コマンド

規格では沢山の FTP コマンドがあり、そのいくつかはファイル・システムの操作に関係するものです。本デザインでは最小限の FTP コマンドのみサポートしており、2つの FTP クライアント・ソフトウェアと接続する前提で実装されています、ひとつは FileZilla でもうひとつは WindowsOS のビルトイン FTP クライアントです。本デザインで実装されている FTP コマンドの一覧を以下に示します。

FTP コマンド	説明	実装されている FTP 応答	データ転送
USER	ユーザー名の認証	331-ユーザー名 OK, パスワード要求	なし
PASS	パスワードの認証	230-ユーザ・ログイン	
FEAT	サーバーに実装されている拡張コマンドのリストを取得する	211-システム状況 (SIZE, MLST, UTF8)	
PASV	パッシブ・モードに移行する	227-パッシブ・モード移行	
TYPE	転送モードの設定	200-要求動作を完了	
NOOP	何もしない		
PWD	作業ディレクトリを取得	257-"PATHNAME" 生成	
XPWD	現在のディレクトリを表示		
SIZE	ファイル・サイズを返す	213-ファイル・ステータス	
QUIT	接続を終了	221-制御コネクションをクローズ	
LIST	指定ファイルまたはディレクトリの情報一覧を返す	150-ファイル・ステータス OK 226-データ・コネクションをクローズ	
MLSD	ディレクトリ内のファイル一覧を返す		
RETR	リモート・ファイルをダウンロードする		
STOR	ファイルをアップロードする		

表 1: サポートする FTP コマンド一覧

表 1 に示すとおり 3 列目は本デザインで実装されている FTP 応答番号を示します。4 列目はそれぞれのコマンドにデータ・コネクションが必要か否かを示します。このデータ・コネクションを必要とするコマンドすなわち LIST, MLSD, RETR, STOR はそのコマンドに先立ってクライアントより PASV コマンドが発行されます。一方その他の FTP コマンドはデータ転送を必要とせずサーバーからクライアントに FTP 応答のみ返送します。

実装されていない FTP コマンドについては DELE(ファイル消去)コマンドについては応答 550(不許可)を、また CWD, SYST, PORT, OPTS, SITE コマンド等については応答 202(実装していない)のいずれか2種類の応答を返送します。

DELE コマンドは実装されていないため、ユーザーはサーバーのファイルを消去することはできません。サーバー内のファイルは DDR3 メモリ内に格納されているため、システムの電源を切断するとサーバー内のファイルは見えなくなります。

3.3 FTP サーバー

FTP サーバー内ファームウェアの制御シーケンスは以下の通りです。

- 1) TOE1G-IP と AXIEMAC の以下のパラメータを初期化します
 - FTP サーバーMAC アドレス = 0x000102030405
 - FTP サーバーIP アドレス = 192.168.11.25
 - FTP クライアント IP アドレス = 192.168.11.42
 - FTP サーバーのデータ・コネクション用ポート番号 = 50000
 - TOE1G-IP コア・タイムアウト時間 = 2 秒
- 2) TOE1G-IP コアの RST を解除し TOE1G-IP コア機能の ARP 要求/応答をサーバーとクライアント間で開始します。CPU は TOE1G-IP コアの BSY フラグが '0' クリアされるのを待機します。
- 3) ファームウェア内のパラメータを初期化します。
- 4) RXADDR_REG をポーリングし RXRAM 内にデータが受信されるのをモニタします。
- 5) 受信パケットを一時バッファにファームウェアで移します。このとき“TMPBUF_SIZE”で指定された 256 バイト・サイズのみが対象となります。
- 6) IP と TCP のチェックサムを計算しパケットが有効であることを確認します。チェックサムが計算値と合致しない場合エラーを返します。
- 7) パケット内のクライアント側ポート番号を現在アクティブなセッションのポート番号と比較します。
 - a) ポート番号値が合致していた場合、アクティブ・セッションのシーケンス番号と ACK 番号をロードし次の TCP パケット転送を継続します。
 - b) 値が合致せず新しいセッションである場合、クライアント側ポート番号とシーケンス番号/ACK 番号の初期値をセットし TCP パケット転送を開始します。

注意: このデモのファームウェアでは新しいセッションを最大 50 までサポートします。またセッションはクローズすると再利用できません。
- 8) FTP 制御ポートにて以下 4 種類のパケットを受信します。
 - a) SYN フラグ: FTP 制御の新しいセッションをオープンするパケットです。ACK を送信するため、CPU は TCP パケットを TxRAM 内に作成し EMAC へ転送します。セッションが確立されると CPU はクライアントに歓迎のメッセージを送信します。
 - b) FIN フラグ: FTP 制御の現在のセッションをクローズするパケットです。SYN フラグと同様 CPU は TCP パケットを作成し ACK を戻します。その後コネクションは解放されます
 - c) データを伴う ACK: 制御ポートにおいてクライアントからのデータは FTP コマンドのみです。表 1 の第 4 列に示すとおり FTP コマンドは以下の 2 種類に分類されます。
 - データ・コネクションを伴わない FTP コマンド、この場合 FTP 応答は TxRAM を通して CPU により作成されます。各 FTP コマンドに対する FTP 応答の詳細については表 1 の第 3 列を参照してください。
 - データ・コネクションを伴う FTP コマンド、この場合 CPU は CONNON_REG レジスタをモニタし、データ・コネクションが確立されるのを待ちます。その後 CPU により応答 150 が返送されサーバー側のデータ・コネクションの準備が整ったことを通知します。データ転送が完了しデータ・コネクションがクローズすると CPU は応答 226 返送しデータ・コネクションがサーバー側から解放されたことを通知します。データ・コネクションを伴う FTP コマンドのシーケンス詳細については次の章で説明します。
 - d) データを伴わない ACK: クライアントからの ACK パケットです。
- 9) ステップ 4)に戻りクライアントからの新しい FTP コマンドを待ちます。

3.4 データ・コネクションを伴う FTP コマンドのシーケンス

3.4.1 LIST

- 1) CONNON_REG レジスタをモニタしデータ・コネクションがすでにオープンされているかを確認します。
- 2) TxRAM に FTP 応答 150 をライトし EMAC へ送信します。
- 3) DDR3 アドレス、転送サイズなどのパラメータをセットし TOE1G-IP コアと AXITOE1G レジスタに転送スタートを指示することでファイル情報エリア内のファイル情報をデータ・コネクション経由で返送します。
- 4) TOE1G_CMD_REG レジスタのビット 0 をモニタし、TOE1G-IP コアが転送を完了するのを待ちます。
- 5) AXITOE1G_CTRL レジスタのビット 0 をモニタし、AXITOE1G が DDR から EMAC へ全データの転送を完了するのを待ちます。
- 6) TOE1G-IP コアのレジスタをセットしデータ・コネクションをアクティブ・クローズします。
- 7) TOE1G_CMD_REG レジスタのビット 0 をモニタし、TOE1G-IP コアがコネクションの解放を完了するのを待ちます。
- 8) TxRAM に FTP 応答 226 をライトし EMAC へ送信します。

3.4.2 MLSD

このコマンドは LIST コマンドと同じシーケンスで実装されています。相違点としては返送するデータを格納する DDR アドレスです。MLSD コマンドではファイル名・サイズ・エリア(FILE_NAME_SIZE_AREA)内のデータを返送します。

3.4.3 STOR

- 1) FTP コマンドからファイル名を抽出します。
- 2) TxRAM に FTP 応答 150 をライトし EMAC へ送信します。
- 3) TOE1G-IP コアの受信 FIFO にデータが到着するのを待ちます、そして AXITOE1G の制御信号をセットし TOE1G-IP コアから DDR3 メモリへデータを転送します。DDR3 の転送先開始アドレスはデータ・エリア(FTP_DATA_AREA)の空きスペースの先頭アドレスとなります。
- 4) 各データ転送の終わりに、次の転送準備のため DDR3 階氏アドレスと全ファイル・サイズを再計算します。
- 5) ステップ 3) – 4) を、クライアントからデータ・コネクションがクローズされるまで繰り返します。コネクションの状況は CONNON_REG レジスタをモニタすることで確認できます。
- 6) TOE1G-IP コアと AXITOE1G モジュールが全データ転送を完了するまで待ちます。
- 7) ファイル情報を更新します、すなわち、ファイル名・サイズ・権限・更新日情報をファイル情報エリア(FILE_INFO_AREA)とファイル名・サイズ・エリア(FILE_NAME_SIZE_AREA)の両方で更新します。ファイル・サイズはステップ 4)で計算された全データ転送サイズと同じになります。さらに、次の空きスペースでの開始アドレスは現在の転送の最終アドレスと一致します、ただし管理を簡単にするためセクタ単位で切り上げます。
- 8) TxRAM に FTP 応答 226 をライトし EMAC へ送信します。

3.4.4 RETR

- 1) LIST コマンドのステップ 1) – 2) と同じです。
- 2) FTP コマンドからファイル名を抽出します、そして合致するファイル名を検索します。
- 3) 合致したファイルのデータを格納する DDR3 の開始アドレスとファイル・サイズをロードし、データのバースト転送を行うためレジスタにセットします。
- 4) LIST コマンドのステップ 3) – 8)と同等の処理を行います、ただしこの RETR コマンドにおける DDR3 の開始アドレスと総転送サイズはステップ 2)と 3)で得たものをセットします。

4 配慮すべき注意事項

本リファレンス・デザインの FTP サーバーのソースコードには違反操作や期待されない操作に対するエラー・リカバリは含まれていません。ユーザーが FTP サーバー側ソフトウェアの動作を理解しやすくするため、ソフトウェアは簡略化されており、ソースコードでは最小限の FTP コマンドのみがサポートされます。実際に使われる FTP コマンドは FTP クライアント側ソフトウェアに依存しますが、本デザインでは一般的な FileZilla と WindowsOS のビルトイン FTP クライアントで必要なコマンドだけが実装されます。またリソースの消費量を最小限に抑えるため、本デザインは 16 ファイルまでしかサポートされず、ファイル名の文字数も最大 30 文字となっています。さらに、ファイル・システムのデザイン簡略化のためファイル消去コマンドは実装されていません。

制御コネクションにおいては、ファームウェアは送信・受信のどちらもパケット・ロスト発生時のデータ・リカバリは行っておりません。さらにファームウェアではパケットから TCP ヘッダ長を抽出する機能も実装しておらず、TCP ヘッダ長は 20 バイト固定を前提としてデザインされています。

本リファレンス・デザインは実用的なものではなく、あくまで TOE1G-IP コアで FTP システムを実装するサンプル・デザインとして参照してください。ただ以下図 10 に示すとおり、TOE1G-IP コアを FTP システムに導入することにより、GbE にて 100M バイト/秒を超えるファイルの転送速度を実現でき、本デザインにて実機で基本動作確認が可能となります。

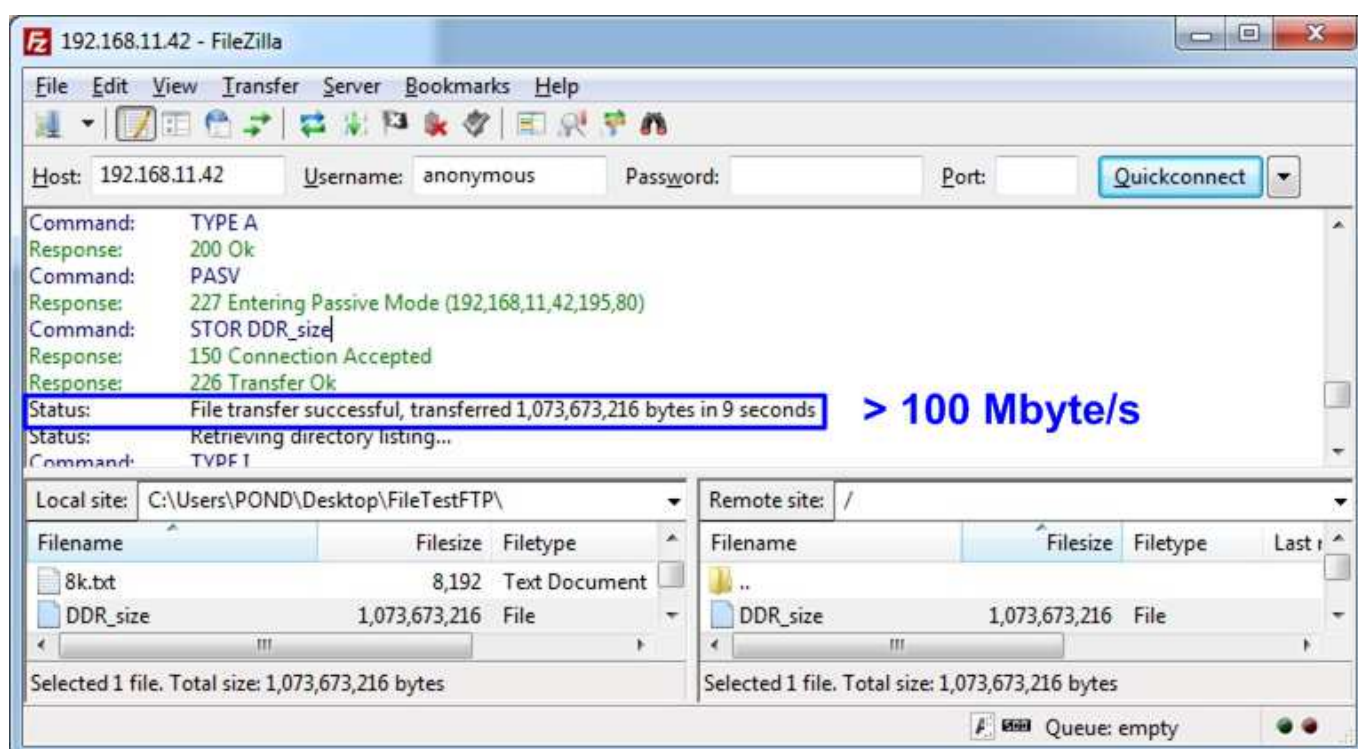


図 10: FileZilla での実機テスト結果例

5 更新履歴

リビジョン	日付	説明
1.0	9-Jan-15	Initial version release
1.0J	2016/03/30	日本語版作成
1.1J	2016/9/1	製品名の変更(TOE2-IP → TOE1G-IP)